AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-9 **(Cancelled)**

10. (New) In a computing system having one or more computers configured to invoke one or more composite software services, each composite software service containing one or more embedded software services, a method of guaranteeing an invocation of a composite software service, the method comprising:

identifying a composite software service containing one or more embedded software services;

receiving a request to invoke the composite software service;

tracking the connectivity and data dependencies among the one or more embedded software services contained within the composite software service;

tracking a context of the composite software service, the context including states of execution of the one or more embedded software services;

associating a unique ID for each of the one or more embedded software services;

invoking one of the one or more embedded software services;

when receiving an indication that invocation of one of the one or more embedded software services has succeeded, updating a state of the successful embedded software service; and

when receiving an indication that invocation of one of the one or more embedded software services was unsuccessful, using the unique ID associated with the unsuccessful embedded software service to re-invoke the unsuccessful embedded software service.

11. (New) The method of claim 10, wherein the one or more embedded software services are nested to any depth within the composite software service.

12. (New) The method of claim 10, wherein the invocation and reinvocation of one of the one or more embedded software services is performed based on one or more guaranteed invocation attributes associated with the composite software service.

13. (New) The method of claim 12, wherein the one or more guaranteed invocation attributes specify whether to apply guaranteed invocation protocols, a number of reinvocations, and a time between reinvocations.

14. (New) The method of claim 10, wherein the one or more guaranteed invocation attributes define overwriting attributes for the composite software service configured to overwrite attributes for the one or more embedded software services.

15. (New) The method of claim 10, wherein receiving an indication that the invocation of one of the one or more embedded software services was unsuccessful comprises receiving indication of at least one of: an unknown state of an embedded software service, an unsuccessful invocation of an embedded software service due to internal system failure, or an unsuccessful invocation of an embedded software service due to external system failure.

16. (New) The method of claim 10, wherein when receiving an indication that the invocation of one of the one or more embedded software services was unsuccessful, using the unique ID associated with the unsuccessful embedded software service to re-invoke the unsuccessful embedded software service comprises:

tracking a number of unsuccessful invocations for the unsuccessful embedded software service; and

tracking a number of unsuccessful invocations for the composite software service associated with the unsuccessful embedded software service.

17. (New) The method of claim 14, wherein when receiving an indication that the invocation of one of the one or more embedded software services was unsuccessful, using the unique ID associated with the unsuccessful embedded software service to re-invoke the unsuccessful embedded software service comprises overriding attributes for the unsuccessful embedded software service with the one or more guaranteed invocation attributes of the composite software service.

18. (New) The method of claim 10, wherein the one or more embedded software services is associated with a service library, wherein invoking one of the one or more embedded software services comprises using the unique ID associated with the invoked embedded software service to access a wrapper interface associated with the service library.

19. (New) In a computing system having one or more computers and performing one or more composite software services, each composite software service containing one or more embedded software services, a method of guaranteeing an invocation of a composite software service, the method comprising:

identifying a composite software service containing one or more embedded software services;

identifying one or more guaranteed invocation attributes associated with the composite software service;

receiving a request to invoke the composite software service;

generating an execution graph representing connectivity and data dependencies among the one or more embedded software services contained within the composite software service;

generating an invocation map having data structures to hold a context of the composite software service, the context including states of execution of the one or more embedded software services;

traversing the execution graph and ordering the one or more embedded software services into steps of invocation;

associating a unique ID for each step of invocation;

preparing input data for each step of invocation;

invoking each step of invocation in the order defined by the execution graph;

when receiving an indication that the invocation of one of the one or more embedded software services has succeeded, recording a state and an output of the successful embedded software service in the invocation map; and

when receiving an indication that the invocation of one of the one or more embedded software services was unsuccessful, using the unique ID associated with the unsuccessful embedded software service to re-invoke the unsuccessful embedded software service.

20. (New) The method of claim 19, wherein the one or more embedded software services are nested to any depth within the composite software service.

21. (New) The method of claim 19, wherein the one or more guaranteed invocation attributes are implemented using a software service interface on the composite software service.

22. (New) The method of claim 19, wherein the one or more guaranteed invocation attributes specify whether to apply guaranteed invocation protocols, a number of reinvocations, and a time between reinvocations.

23. (New) The method of claim 19, wherein the one or more guaranteed invocation attributes define overwriting attributes for the composite software service configured to overwrite attributes for the one or more embedded software services.

24. (New) The method of claim 19, wherein the invocation map holds the context of the composite software service in a persistent context mechanism implemented as a set of service interfaces that provides a layer of encapsulation to an underlying storage medium.

25. (New) The method of claim 19, further comprises at least one of:

wherein preparing input data for each step of invocation comprises storing the input data in the invocation map;

wherein invoking each step of invocation in the order defined by the execution graph comprises logging such attempt in the invocation map; or

wherein associating a unique ID for each step of invocation comprises storing the unique ID in the invocation map.

26. (New) The method of claim 19, wherein receiving an indication that the invocation of one of the one or more embedded software services was unsuccessful comprises receiving indication of at least one of: an unknown state of one of the one or more embedded software services, an unsuccessful invocation of an embedded software service due to internal system failure, or an unsuccessful invocation of an embedded software service due to external system failure.

27. (New) The method of claim 19, wherein when receiving an indication that the invocation of one of the one or more embedded software services was unsuccessful, using the unique ID associated with the unsuccessful embedded software service to re-invoke the unsuccessful embedded software service comprises:

incrementing an unsuccessful invocation retry counter for the unsuccessful embedded software service; and

incrementing an unsuccessful invocation retry counter for the composite software service associated with the unsuccessful embedded software service.

28. (New) The method of claim 23, wherein when receiving an indication that the invocation of one of the one or more embedded software services was unsuccessful, using the unique ID associated with the unsuccessful embedded software service to re-invoke the unsuccessful embedded software service comprises overriding attributes for the unsuccessful embedded software service with the one or more guaranteed invocation attributes of the composite software service.

29. (New) The method of claim 19, further comprising:

identifying a second composite software service embedded within the composite software service; and

generating a second invocation map embedded within the invocation map for the composite service, the second invocation map having data structures to hold a context of the second composite software service.

30. (New) The method of claim 19, wherein the one or more embedded software services is associated with a service library, wherein invoking one of the one or more embedded software services comprises using the unique ID associated with the invoked embedded software service to access a wrapper interface associated with the service library.